Chetana's

Hazarimal Somani College of Comm. & Eco.

Smt. Kusumtai Chaudhari College of Arts

(Autonomous)

(Affiliated to University of Mumbai)

Bandra (East), Mumbai - 400 051.

MAHARASHTRA

2024-2025

"VIDEOTUBE WEB-APP"

A PROJECT REPORT

Submitted in partial fulfillment

of the Requirements for the Degree of

BACHELOR OF SCIENCE

(INFORMATION TECHNOLOGY)

PROJECT GUIDE

MR. SARAVANAN REDDY

SUBMITTED BY

SAHIL MANE: - 338

Proforma for the approval project proposal

PRN No: 2022016402594123 **Roll no: 338**

1) **Name of the Student:**
   - Sahil Manoj Mane
2) **Title of the Project:** VideoTube Web App
3) **Name of the Guide:** Prof. Saravanan Reddy
4) **Teaching experience of the Guide:** 22 years
5) **Is this your first submission?** Yes No

Name & Signature of member: -

SAHIL MANE

Date…….

Signature of the External

Date…….

Signature of the Internal

Date…….

ABSTRACT

The "VideoTube" web application is a robust platform designed for video hosting and community interaction, inspired by YouTube. It addresses the challenges of managing video content and user engagement with a scalable and efficient solution.

The application is built using a service-oriented architecture (**SOA**) and a decoupled design, employing the MERN stack for its implementation. MongoDB Atlas is used for the database, ensuring scalability and efficient data management. Express.js handles backend logic and API endpoints, while React.js powers the frontend, offering a responsive and dynamic user interface. Node.js serves as the runtime environment, facilitating server-side processing.

The platform integrates with Cloudinary to manage and scale video storage, user avatars, and thumbnails, enhancing media management and performance.

Features:

- **User Authentication**: Provides secure login and registration with OAuth support for third-party integrations.
- **Video Hosting**: Allows users to upload, manage, and stream videos on their personal channels.
- Community Interaction: Supports tweet-like posts, comments, likes, and dislikes for videos and community updates.
- **Responsive Design**: Ensures a seamless user experience across various devices and screen sizes.
- **API Integration**: Connects the frontend and backend through robust APIs, facilitating smooth data handling and interaction.

The frontend is developed with React.js, offering a dynamic and interactive user interface designed using Figma for a visually appealing experience. The backend, built with Express.js, includes APIs for core functionalities such as user authentication, video processing, community management, and commenting. These APIs are rigorously tested with Postman to ensure reliability and performance.

Scopes: The application provides secure user authentication, video management, community engagement, responsive design, and API integration.
Limitations: The project faces challenges such as resource-intensive video hosting, scalability issues, content moderation needs, data privacy concerns, and technical complexity.
Conclusion:

"VideoTube" leverages modern web technologies and cloud services to deliver a scalable, secure, and user-friendly platform for video hosting and community interaction. By utilizing a decoupled architecture and specialized services, the application aims to provide an optimal user experience and foster a vibrant, interactive community.

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Saravanan Reddy for his guidance and constant supervision as well as for providing necessary information regarding the project and for their support in completing the project.

We would like to express our gratitude towards our parents and members of Chetana's Hazarimal Somani College of Comm. & Eco. Smt. Kusumtai Chaudhari College of Arts, for their kind cooperation and encouragement which helped me in completion of this project.

DECLARATION

We hereby declare that the project entitled, "VideoTube Web-Application" done at Chetana's Hazarimal Somani College of Comm. & Eco. Smt. Kusumtai Chaudhari College of Arts has not been in any case duplicated to submit to any other university for the award of any degree. To the best of our knowledge other than us, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as a final semester project as part of our curriculum.

Name & Signature of members: -

Sahil Mane

Table of contents

CHAPTER 1: INTRODUCTION

The "VideoTube" web application is designed to enhance video hosting and community interaction, providing an efficient and scalable solution for managing digital content and user engagement. This platform addresses the challenges associated with traditional video management systems, which can be cumbersome and time-consuming.

System Overview:

The "VideoTube" application is a web-based platform aimed at offering a seamless experience for video hosting and community engagement. The system includes several key modules:

- **User Authentication**: This feature allows users to securely log in and register using OAuth integration, enabling access across multiple devices while maintaining high security.
- **Video Management**: Users can upload, manage, and stream videos on their personal channels. The system supports the creation of individual channels for users, providing a dedicated space for content.
- **Community Interaction**: The application includes features for community engagement, such as tweet-like posts, comments, likes, and dislikes. This encourages interaction and participation within the platform, similar to the community tab found on YouTube.
- **User Authentication and Video Management**: Users can securely log in, upload videos, and manage their channels from any device.
- **Community Features**: Users can engage with each other through posts, comments, and reactions, promoting a vibrant and interactive community.
- **Responsive Interface**: The application adapts to various devices and screen sizes, providing a consistent user experience.
- **Efficient Data Handling**: APIs facilitate seamless communication between the frontend and backend, supporting real-time updates and interactions.

1.1 BACKGROUND

- **Manual Video Management**: Traditional systems rely on manual processes for uploading and managing video content, which are time-consuming and error-prone.
- **Inefficiencies in Data Handling**: Manual updates and record-keeping for videos and user interactions can lead to inconsistencies and increased workload.
- **Risk of Data Loss**: Older systems are prone to issues such as data loss, unauthorized alterations, and difficulty in tracking changes.
- **Scalability Issues**: Handling large volumes of videos and user interactions manually becomes impractical as the platform grows.
- **Limited User Engagement Features**: Traditional methods lack advanced tools for managing and engaging with community interactions effectively.

1.2 OBJECTIVES:

The primary objective of the "VideoTube" web application is to modernize and streamline video hosting and community interaction by replacing outdated and inefficient manual methods with a secure and optimized digital solution.

- **Efficient Video Management**: The application replaces manual video management processes by allowing users to upload, manage, and stream videos directly through an intuitive web interface.
- **Automated Data Handling**: It automates the management and organization of video and user data, eliminating the need for manual updates and ensuring efficient data handling.
- **Seamless Community Engagement**: The system simplifies community interaction by providing automated features for posting updates, commenting, and managing likes and dislikes, reducing manual effort.
- **Enhanced User Experience**: Incorporates a responsive design with React.js and Cloudinary integration to ensure a seamless and engaging user experience across various devices.
- **Robust API Integration**: Utilizes well-tested APIs for reliable communication between the frontend and backend, ensuring smooth operation and real-time updates.

1.3 PURPOSE:

a. Web-based video management system:
The "VideoTube" application provides a modern solution for video hosting and community interaction, offering a more efficient, convenient, and secure alternative to traditional methods.

b. **Remote Access:**
Users can manage their video content and community interactions from anywhere, with the ability to upload videos, engage with content, and track activity without being tied to a specific location.

c. User-Friendly Interface:
Designed with Tailwind CSS and Figma, the system features a streamlined, intuitive user interface that simplifies the management of videos and community interactions. It includes a chatbot for handling common tasks, reducing manual effort.

d. Security:
The application is securely integrated with MongoDB Atlas and Cloudinary, providing robust data protection and authorization features. It minimizes risks such as data loss or unauthorized access by enforcing role-based access controls.

e. Data-Driven:
By leveraging cloud-based storage and database management, the application eliminates the need for physical files or sheets. Users can manage and track video and interaction data directly through the platform.

f. Efficiency:
The system enhances productivity by automating tasks such as video management and community engagement, allowing users to accomplish more in less time. The chatbot feature further streamlines daily operations with minimal commands.

1.4 SCOPE:

- **Comprehensive Video Management**: The "VideoTube" application manages and tracks video content and community interactions over time, generating reports and data for a large user base.
- **Role-Based Access**: Users, including content creators, viewers, and administrators, can access the platform based on their roles and permissions, ensuring appropriate access and functionality.
- **Automated Engagement Reports**: The system automatically generates reports on user engagement and community interactions, providing insights at regular intervals.
- **Chatbot Assistance**: A chatbot feature is integrated to handle common tasks and queries, enhancing user convenience and interaction efficiency.
- **Data Management and Export**: The application supports automatic data management and can generate reports in various formats, including Excel, for analysis and record-keeping.

1.5 APPLICABILITY:

a. SAVE TIME:

- The "VideoTube" application significantly reduces manual work by automating video management, content interaction, and report generation. This efficiency allows users to focus more on creating and engaging with content rather than handling administrative tasks.

b. MORE CONVENIENT:

- By automating tasks such as video uploads, community interactions, and report generation, the application enhances convenience and efficiency. Users benefit from a streamlined experience with minimal effort required. The integrated chatbot assists with routine tasks and queries, further simplifying operations.

c. SECURED:

- Unlike traditional methods of managing video content and user data, which can be prone to security issues, the application ensures secure data handling through integration with MongoDB Atlas and Cloudinary. Data is stored securely, with access controls in place, and can be exported in various formats, including Excel, for ease of use and further analysis.

1.6 ACHIEVEMENT:

a. **Enhanced User Engagement:**

- The application facilitates increased user interaction through features such as video uploads, community posts, likes, and comments, which drive greater engagement and participation.

b. **Seamless Video Management**:
    - With integrated video hosting and management, users can easily upload, organize, and stream videos on their personal channels. This streamlined process simplifies content management and enhances user experience.

c. Efficient Community Interaction:
    - The inclusion of community features similar to social media allows users to interact through posts, comments, and likes, fostering a vibrant and active user community.

d. Scalable Infrastructure:
    - By leveraging cloud services like MongoDB Atlas for database management and Cloudinary for media storage, the application ensures scalability and high performance, accommodating growing amounts of user-generated content.

e. Real-Time Analytics:
    - The application provides real-time data on video performance and user interactions, enabling content creators and administrators to make informed decisions and adjust strategies quickly.

f. Secure and Reliable Platform:
    - With robust authentication mechanisms and secure data storage, the application ensures user data privacy and platform integrity, addressing concerns about data security and reliability

CHAPTER 2: SURVEY OF TECHNOLOGY

2.1 Existing Systems.

The existing video hosting and community interaction platforms can be categorized into several types:

- **Basic Video Upload and Playback Systems**:
  Platforms that allow users to upload videos and provide basic playback functionality. These systems typically do not support advanced features like community interactions, personalized channels, or content monetization. They are limited in scope and often lack scalability and user engagement tools.
- **Social Media-Based Video Sharing:**
  Platforms like Facebook and Instagram that allow users to share short videos within a social network. While these platforms offer community interaction features like comments and likes, they are not optimized for hosting longer, high-quality videos or building dedicated content channels.
- **Traditional Content Management Systems (CMS):**
  Websites and platforms using traditional CMS to manage and display video content. These systems often involve manual content management processes, are less scalable, and lack the dynamic, user-driven interactions that modern video platforms require.
- **Subscription-Based Video Services:**
  Platforms like Netflix or Hulu that focus on providing curated content through subscription models. These systems are robust for content delivery but do not offer user-generated content or community interaction features, limiting user engagement to passive consumption.
- **Legacy Video Hosting Services:**
  Older video hosting platforms like Vimeo or Dailymotion that provide video upload and sharing features but may lack advanced community interaction tools, scalability, and integration with modern cloud services. These platforms can be restrictive in terms of content distribution and audience engagement.

2.2 List of Technologies Selected

a. **Code Editor:** VS Code
b. **Designing:** Figma.
c. **Preprocessors:** PostCSS.
d. **Front-end:** JSX, CSS.
e. **Back-end:** JS.
f. **Runtime:** NodeJS
g. **Frameworks/Libraries:**
- **Frontend:** ReactJS, NextJs, Tailwind CSS
- **Backend:** ExpressJs

g. Dependencies: Git, GitHub, Postman, NPM
h. **Database:** MongoDB.

2.3 Reason for Selected Technology

- **Challenges with Existing Systems:** As outlined in Section 2.1, existing video hosting and community interaction platforms often lack scalability, advanced community features, and the flexibility needed for modern content creators. Some systems are outdated, less secure, or costly to maintain, while others struggle with user engagement and content management.
- **Technology Selection Rationale:** We have chosen a modern, JavaScript-driven tech stack that is well-suited for building scalable, interactive, and dynamic web applications. The primary reason for selecting these technologies is their ability to support a decoupled architecture and service-oriented approach, which aligns with the project's goals of scalability, performance, and user engagement.
    - **JavaScript Ecosystem:** JavaScript is a versatile language that allows for both frontend and backend development. This unification simplifies the development process and ensures a consistent and efficient workflow across the entire application.
    - **React.js for Frontend:** React.js is a powerful JavaScript library for building user interfaces, particularly single-page applications. It enables the creation of a dynamic, responsive, and interactive frontend, providing users with an optimized and seamless experience across devices.
    - **Express.js for Backend:** Express.js, a minimal and flexible Node.js web application framework, is used for handling server-side logic, routing, and API development. It ensures that the backend is lightweight, maintainable, and capable of supporting high traffic loads.
    - **MongoDB Atlas for Database:** MongoDB Atlas is a cloud-based document database that simplifies the storage of structured and unstructured data. Its flexible schema design and scalability make it ideal for managing video content, user profiles, and community interactions in a distributed environment.
    - **Cloudinary for Media Management:** Cloudinary is integrated for efficient media storage and management, handling video uploads, user avatars, and thumbnails. It ensures fast delivery and optimal performance, crucial for a video-centric platform.
    - **Service-Oriented Architecture (SOA):** The application is designed with a decoupled architecture, using SOA principles to ensure that different components, such as video processing, user authentication, and community interactions, are modular and can be developed, tested, and scaled independently.

These technologies were selected to build a robust, scalable, and secure platform that meets the demands of modern content creators and users, ensuring a high-quality user experience and efficient system performance.

a. Code Editor

1. **Visual Studio Code**
    - Visual Studio Code is a lightweight but powerful source code editor that runs on your desktop and is available for Windows, macOS, and Linux.
    - It comes with built-in support for JavaScript, TypeScript, and Node.js and has a rich ecosystem of extensions for other languages (such as PHP, Python, HTML, CSS, etc.) and runtimes (such as NPM, Node, Express, json server).
    - Aside from the whole idea of being lightweight and starting quickly, VS

Code has IntelliSense code completion for variables, methods, and imported modules; graphical debugging; linting, multi-cursor editing, parameter hints, and other powerful editing features; snazzy code navigation and refactoring; and built-in source code control including Git support. Much of this was adapted from Visual Studio technology.

b. Designing

1. Figma

- The name Figma comes from the fact that it's the first interface design tool with real-time collaboration – or, as we like to say, it's a "figment of your imagination".
- PRO TIP: Figma is a vector graphics editor and prototyping tool it is used by designers to create high-quality designs and prototypes.

c. Modeling

1. Eraser.io

- Eraser.io is a web-based modeling tool that simplifies the process of creating, sharing, and collaborating on system designs and diagrams.
- It supports a variety of diagram types including flowcharts, UML diagrams, and ER diagrams, making it versatile for modeling different aspects of the "VideoTube" web application.
- The tool is particularly useful for collaborative modeling, allowing team members to work together in real-time, which is crucial for maintaining consistent and up-to-date models.
- Eraser.io is designed with simplicity and accessibility in mind, offering an intuitive interface that requires minimal setup, making it ideal for both technical and non-technical stakeholders involved in the project.

2. Draw.io

- Draw.io is a versatile, web-based diagramming tool that enables users to create a wide range of diagrams, including UML, ER diagrams, flowcharts, and more.
- It offers extensive template options and a user-friendly interface, making it easy to visualize and design various components of the "VideoTube" web application.
- The tool is highly collaborative, allowing team members to work on diagrams together in real-time, which is essential for keeping the project's design documents aligned and up to date.
- Draw.io integrates seamlessly with cloud services like Google Drive and OneDrive, enabling easy sharing and storage of diagrams, which is crucial for accessibility and collaboration among all project stakeholders.

3. Excalidraw

- Excalidraw is a lightweight, open-source tool designed for creating hand-drawn style diagrams, making it perfect for quick sketching and brainstorming sessions.
- It offers a simple, intuitive interface that allows for fast diagram creation without the need for extensive setup or technical expertise.
- Excalidraw is especially useful for visualizing ideas and workflows in a more informal and flexible manner, which can be helpful during the initial stages of the "VideoTube" project's design process.
- The tool also supports collaborative drawing, enabling team members to co-create and refine diagrams in real-time, fostering creativity and teamwork.

d. Preprocessors

1. PostCSS

- PostCSS is a powerful tool for transforming CSS with JavaScript plugins. It provides a flexible, modular approach to writing and optimizing CSS, making it an essential part of modern web development workflows.
- It allows developers to use future CSS features today by converting modern CSS into something most browsers can understand, ensuring cross-browser compatibility and enhancing the styling capabilities of your project.
- PostCSS also supports a wide range of plugins, such as autoprefixer (which adds vendor prefixes automatically) and cssnano (which minifies CSS), making the development process more efficient and the final output more optimized.
- Its modular nature means that you can tailor the preprocessing pipeline to suit the specific needs of your "VideoTube" application, ensuring that the CSS is well-optimized and maintainable.

e. Front-end

1. JSX

- JSX is a syntax extension for JavaScript, commonly used with React.js. It allows you to write HTML elements within JavaScript, making the code more readable and easier to maintain.
- It enhances the structure of your application by combining logic and layout within the same file, promoting a seamless development process.

2. CSS (Cascading Style Sheets)

- CSS is used to style and layout your web pages. In conjunction with JSX and React.js, it allows for precise control over the look and feel of the application, including colors, fonts, spacing, and overall layout.
- By using CSS, you can ensure that your application is responsive, visually appealing, and consistent across different devices and screen sizes.

f. Back-end

1. JavaScript

- One of the core technologies of the World Wide Web is JavaScript (JS), a computer language that runs on HTML and CSS.
- JavaScript is used across both the frontend and backend of your application. In the frontend, it powers the dynamic behaviors and interactions within your React.js components.
- A dedicated JavaScript engine is available in all major web browsers for executing code on users' devices. JavaScript is an ECMAScript-compliant high-level, typically just-in-time compiled language.
- Dynamic typing, prototype-based object orientation, and first-class functions are all available.
- For the backend, JavaScript (running on Node.js) manages server-side logic, APIs, and communication with the database.
- As a core web technology, JavaScript is essential for creating interactive and responsive user interfaces.

2. NodeJS

- Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!
- Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.
- A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.
- When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.
- This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

g. Frameworks/Libraries

1. ExpressJS

- Express is a Node.js web application framework that offers a comprehensive collection of functionalities for developing online and mobile apps.
- It allows for the quick creation of Node-based Web applications. Some of the key elements of the Express framework are as follows: It Enables middleware to reply to HTTP requests.
- Defines a routing table that is used to conduct various activities based on a set of criteria through HTTP method and the URL Enables dynamic rendering of HTML pages based on parameters given in the form of templates.

2. Next.js

- Next.js is a React framework that enables server-side rendering (SSR) and static site generation (SSG) for React applications.
- It provides automatic code splitting, optimizing performance by loading

only the necessary JavaScript for each page.

- Next.js supports API routes, allowing you to build backend functionality within the same application.
- It includes built-in support for CSS and Sass, as well as a plugin system for adding additional functionality.
- The framework offers static exporting, enabling you to generate a static version of your site for deployment on any static hosting service.

h. Dependencies

1. Git

- Git is a DevOps tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently.
- Git is used to track changes in the source code, enabling multiple developers to work together on non-linear development.

2. GitHub

- GitHub is the place for open source. With so many great tools available to developers, GitHub has become the place to be for open-source software.
- Some of the biggest open-source projects are hosted on GitHub, such as Ruby on Rails, ReactJs, Bootstrap and many more.

3. Postman

- Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.
- Postman provides an all-in-one platform for API development, allowing you to create, test, and debug APIs efficiently.
- It streamlines the API testing process with an intuitive interface, enabling you to send requests, inspect responses, and automate tests without switching to another tool.

4. **NPM (Node Package Manager)**
    - The Node JavaScript platform's package manager is called Npm [6]. It installs modules and intelligently handles dependency conflicts so that node can locate them.
    - The largest software repository in the world is npm. Many corporations use npm to handle private development, while open-source developers from every continent use it to exchange and borrow packages. we have used nodemailer, bcrypt, express, mongoose, etc.

i. Databases

1. **MongoDB**
    - The term "no NoSQL" is also used to describe systems that do not use SQL.
    - The SQL database (formerly "non-SQL" or "non-relational") provides a system for storing and retrieving data that is not structured using the tabular relations used in relational databases.
    - MongoDB is a document-oriented database that runs on a variety of platforms and offers high performance, high availability, and simple scaling. MongoDB operates on the collection and document concepts.

j. Libraries

1. **ReactJs**
    - ReactJs is a popular JavaScript library for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components that manage their own state, leading to a more efficient and organized codebase.
    - React's component-based architecture makes it easier to build and maintain large-scale web applications, enabling the development of dynamic and responsive user experiences.
    - With React.js, the front end of the "VideoTube" project is highly interactive, allowing for smooth transitions, real-time updates, and efficient handling of complex UI states.
2. **React-Router**
    - React Router is a library for routing in React applications, enabling navigation between different views or pages.
    - It provides components like Route and Link to manage URL routing and navigation within a single-page application.
    - React Router supports nested routes, allowing for complex routing structures and dynamic route matching.
    - It includes features like lazy loading of components to optimize performance and reduce initial load times.
    - The library integrates seamlessly with React’s component-based architecture, maintaining consistent state and UI updates across route changes.

k. Plugins

The main advantage of using plug-ins is the ability to expand the functionality of your website quickly and easily. Webmasters can usually download and install them within minutes. Developers also update plug-ins frequently, sometimes several times year, as they make performance and security improvements.

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 PROBLEM DEFINITION:

- **MANUAL VIDEO MANAGEMENT:**

  Managing video uploads, storage, and streaming manually requires significant effort, including organizing files, setting up server configurations, and handling different media formats, which is inefficient for a growing platform.

- **SCALABILITY ISSUES:**

  Traditional video hosting methods are time-consuming and can struggle with scalability, especially as the number of users and video content grows, leading to performance bottlenecks.

- **DATA SECURITY CONCERNS:**

  Without a secure, automated system, video data, user information, and content integrity are at risk of being compromised, potentially leading to unauthorized access or data breaches.

- **LIMITED USER INTERACTION:**

  Previous systems may not have provided robust community interaction features, limiting user engagement through likes, comments, or community posts, which are essential for building an active user base.

3.2 REQUIREMENT & ANALYSIS:

- The "VideoTube" web application is a live project initiated on [insert start date], aiming to provide a comprehensive platform for video hosting and community interaction.
- The purpose of this web application is to deliver a scalable, secure, and user-friendly solution for managing video content and engaging with users.
- "VideoTube" consists of the following modules:
  - A user-friendly interface for seamless video uploading and channel management.
  - Users can create and manage their own channels, upload videos, and organize content efficiently.
  - Supports community interaction through features like comments, likes, dislikes, and community posts, enhancing user engagement.
  -

3.3 PLANNING AND SCHEDULING:

- The Idea of this project came to me when I thought one day that how complex would be the backend of YouTube, then I thought why think so much when you can do it yourself. And that is how I decided to make this web application and put it in my TY's Final Project.

- I started making the logic of the project and afterwards decided MERN stack as the perfect tech stack for my project.

3.4 S/W & H/W REQUIREMENTS:

a. Software:

Browsers	Chrome, Mozilla, Brave, Edge.
OS	Windows 7 or more.
Dependencies	JavaScript Engine.

b. Hardware (Minimum):

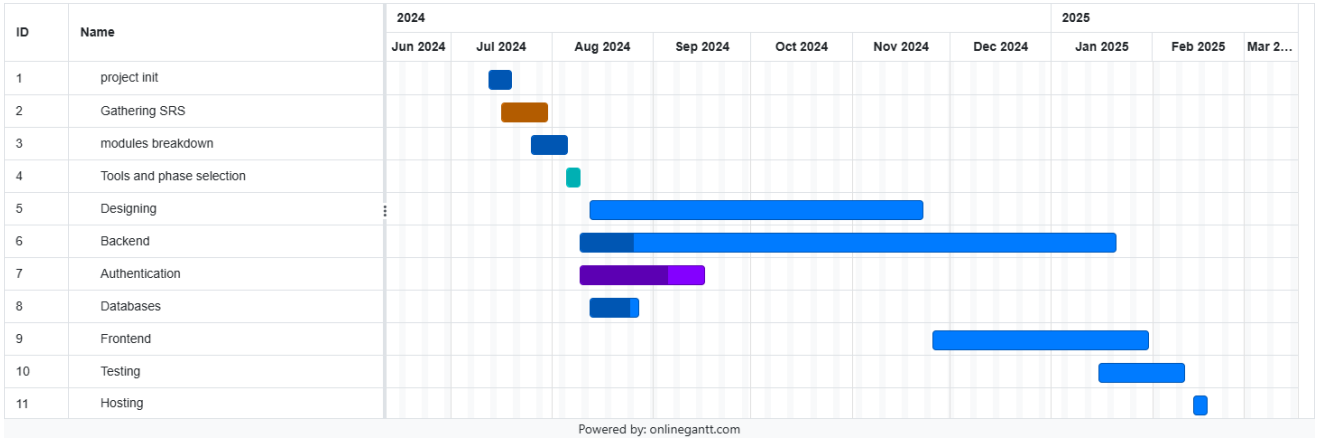
Processor	Duo Core
Ram	2GB
OS architecture	32 bits
Connectivity required	2 Mbps

c. Languages and Frameworks:

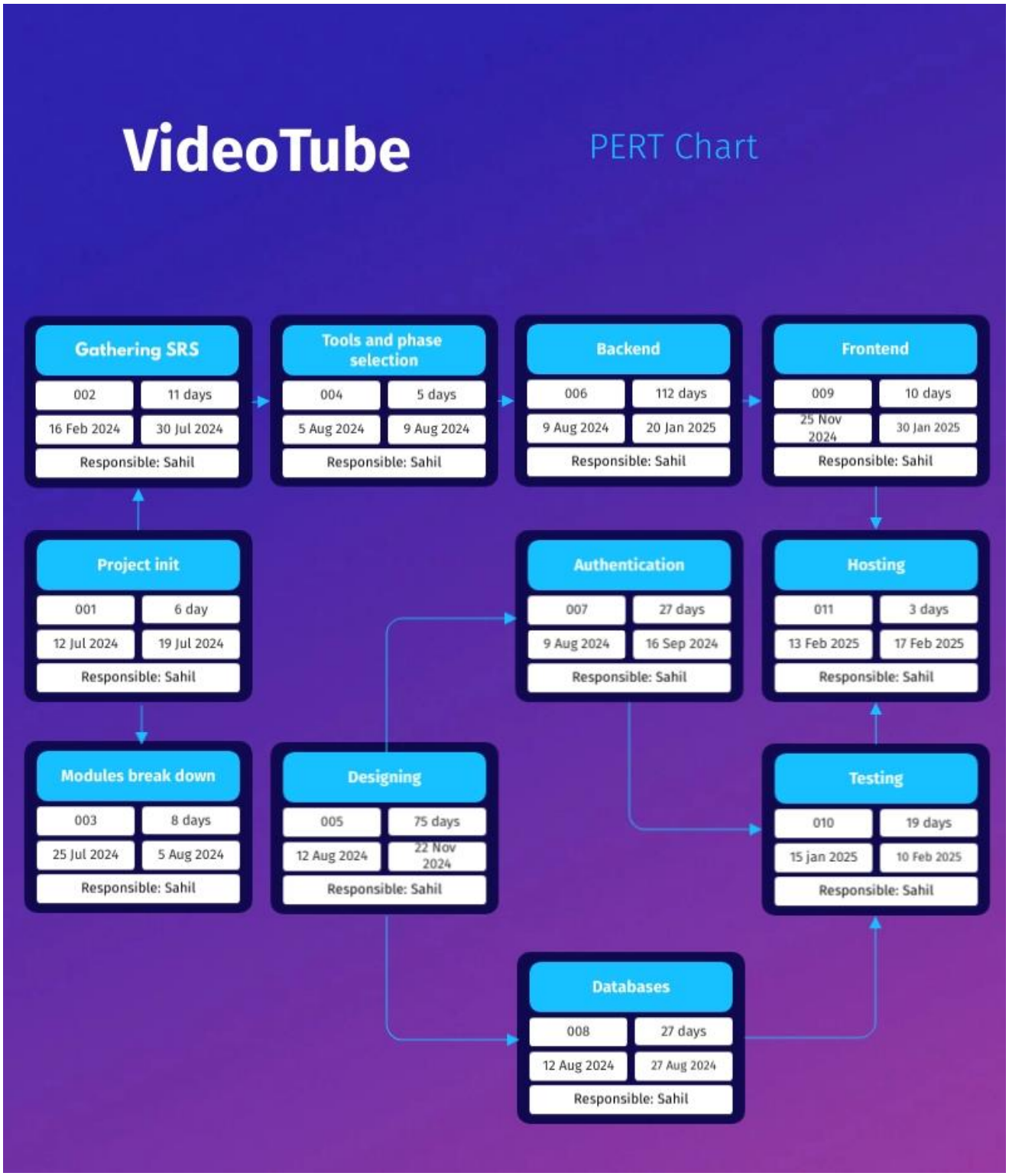
Designing	Figma
System Modeling	Eraser.io, Draw.io, Excalidraw
Front-End	JSX, CSS
Preprocessors	PostCSS
Back-End	JS

Dependencies	Postman, NPM, Git, GitHub, NodeJS
Frameworks	NextJS, ExpressJS
Database	MongoDB
Libraries	ReactJs, React-Router
Plugins	Vite, Google Fonts.

3.5 GANTT CHART:



3.6 PERT CHART:

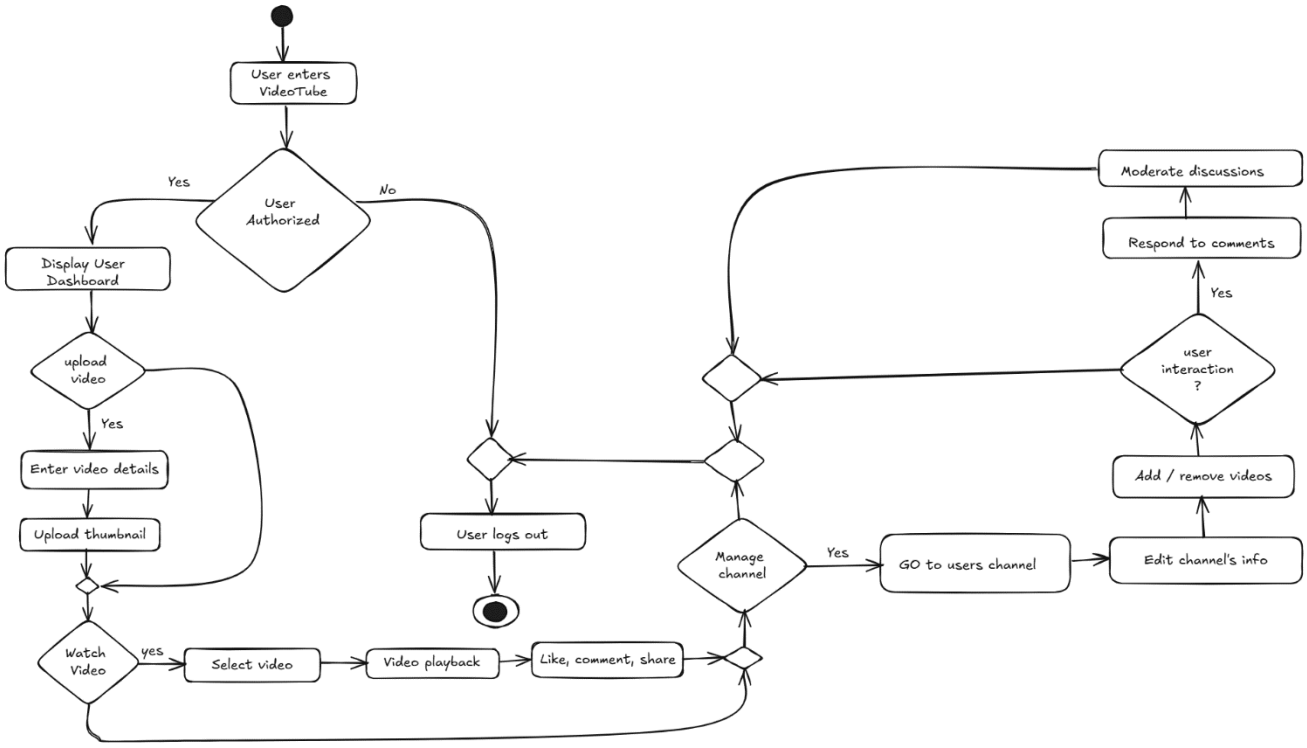

VideoTube
PERT Chart
Gathering SRS
002
11 days
16 Feb 2024
30 Jul 2024
Responsible: Sahil
Tools and phase selection
004
5 days
5 Aug 2024
9 Aug 2024
Responsible: Sahil
Backend
006
112 days
9 Aug 2024
20 Jan 2025
Responsible: Sahil
Frontend
009
10 days
25 Nov 2024
30 Jan 2025
Responsible: Sahil
Project init
001
6 day
12 Jul 2024
19 Jul 2024
Responsible: Sahil
Authentication
007
27 days
9 Aug 2024
16 Sep 2024
Responsible: Sahil
Hosting
011
3 days
13 Feb 2025
17 Feb 2025
Responsible: Sahil
Modules break down
003
8 days
25 Jul 2024
5 Aug 2024
Responsible: Sahil
Designing
005
75 days
12 Aug 2024
22 Nov 2024
Responsible: Sahil
Testing
010
19 days
15 jan 2025
10 Feb 2025
Responsible: Sahil
Databases
008
27 days
12 Aug 2024
27 Aug 2024
Responsible: Sahil

CHAPTER 4: SYSTEM DESIGN

4.1 Conceptual Models

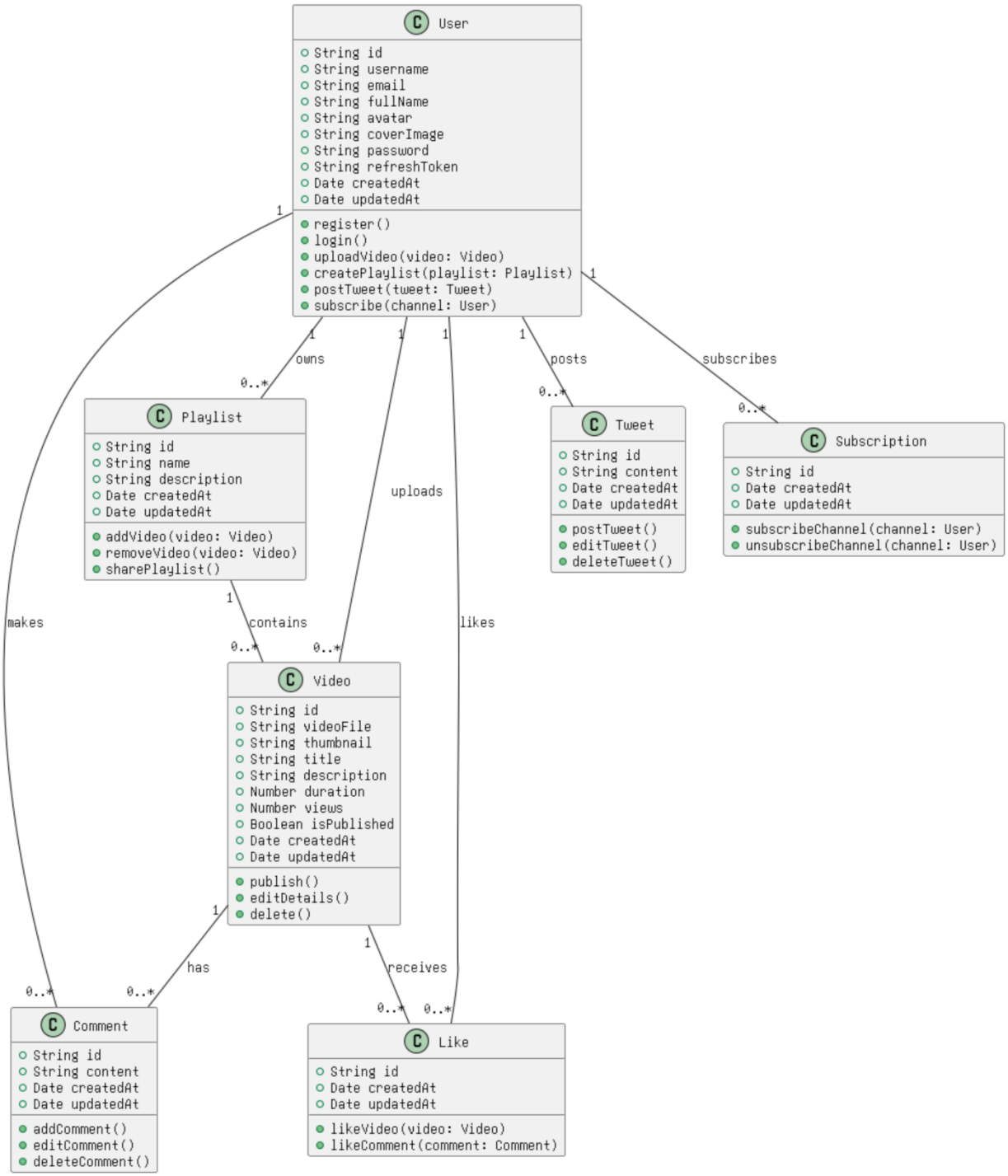
a. ACTIVITY Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with supports for choice, iteration and concurrency. In the Unified Modeling language, activity diagrams are intended to model both computational and organizational processes as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of the application.



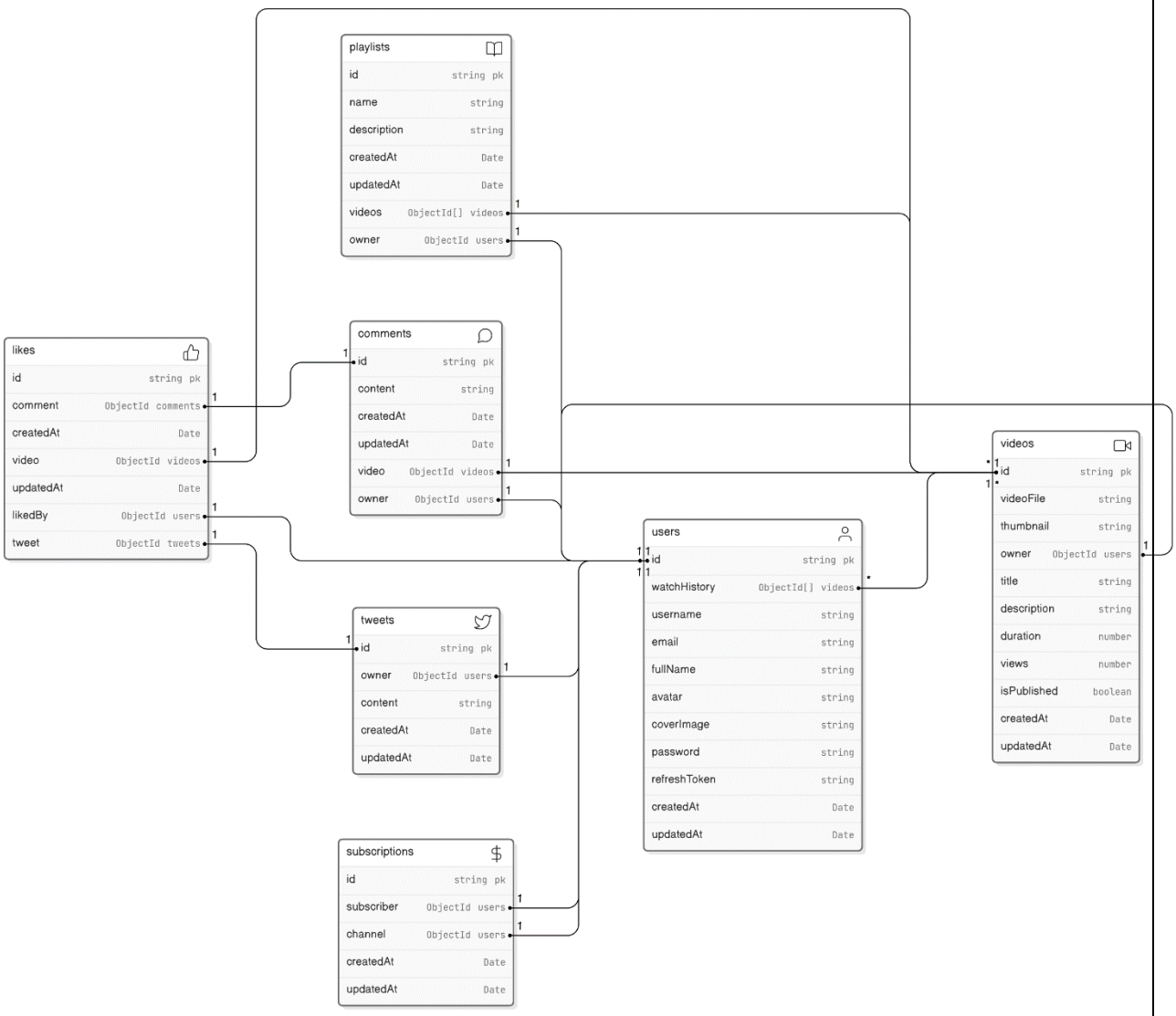
a. Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attribution and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object- oriented languages



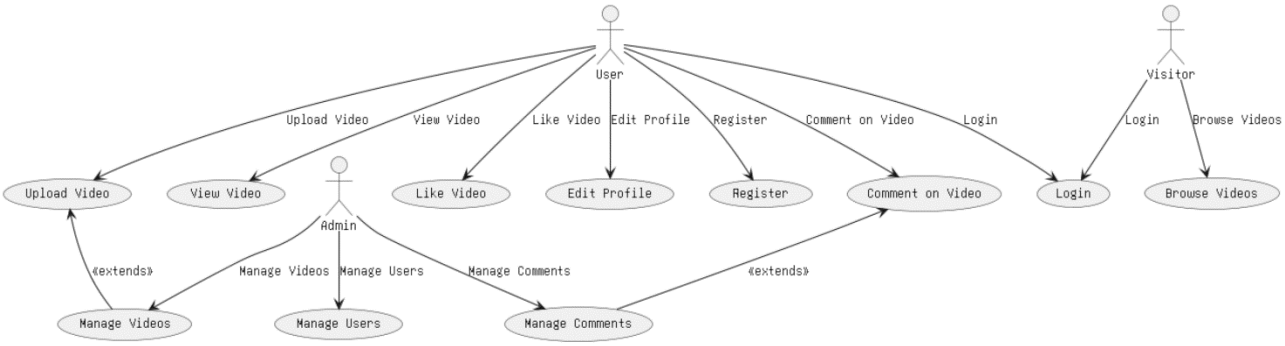
b. ER Diagram:

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationships of entity sets stored in a database. In other words, ER diagrams helps to explain the logical structure of databases. ER diagrams are created based on the basic concepts: entities attributes and relationships. ER Diagrams contains different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.



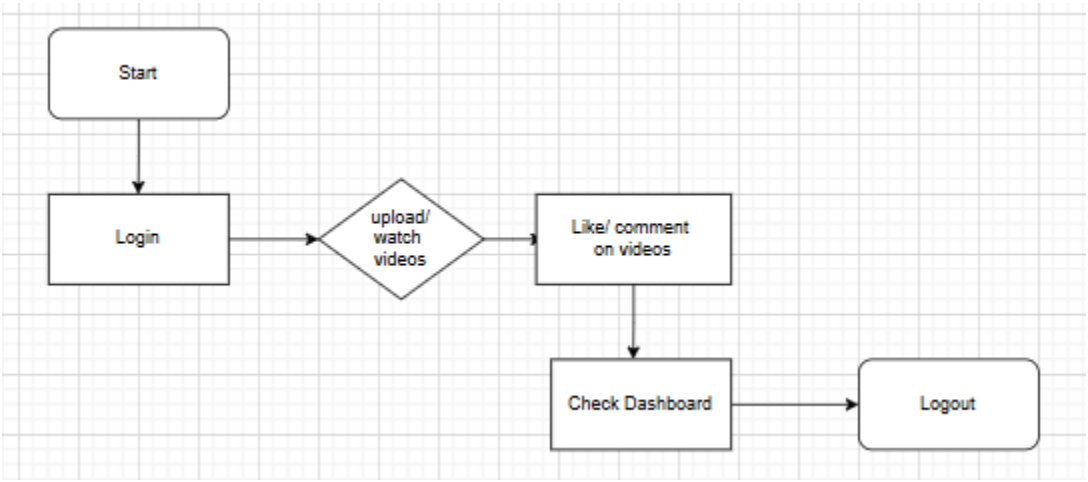
c. Use Case:

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interaction with the system. To build one, you will use a set of specialized symbols connectors. An effective use case diagram can help your team discuss and represent.



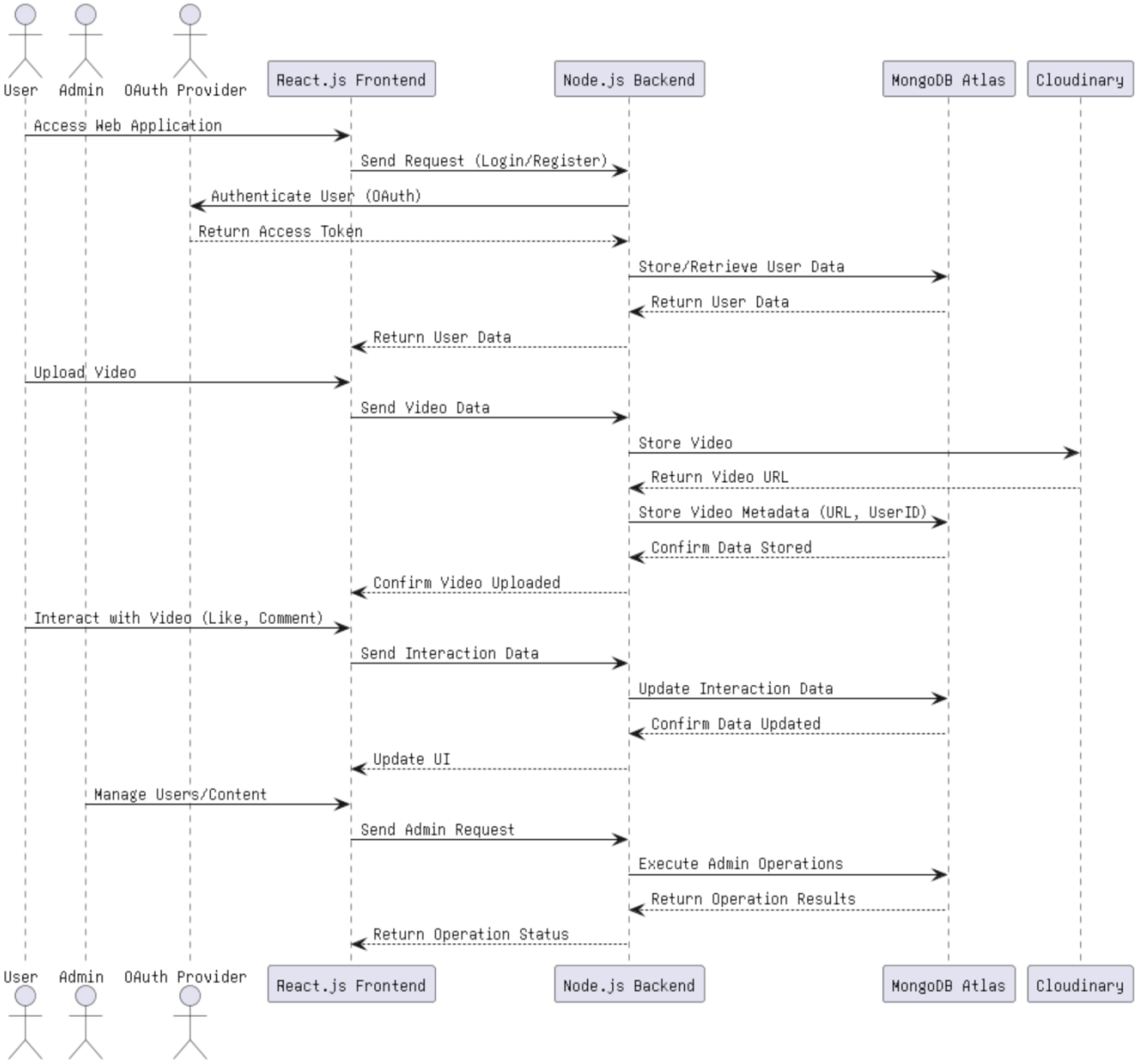
d. Flowchart:

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purpose, and can be used to described various processes, such as a manufacturing process, an administrative or service process, or a project plan.



e. Sequence Diagram:

A Sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their "Lifelines", and the messages that they exchange arranged in time sequence. It does not show the associations among the objects. It represents an Interaction, which is a set of messages exchanged among objects within a collaboration to affect a desired operation or result



4.2 Basic Modules

a. Registration

- **Admin:** Admins will handle the registration of all users on the platform.
- **Content Creators:** Content creators can sign up on the platform or be registered by admins.
- **Moderators:** Moderators can be registered and assigned roles by the admin.
- **Viewers:** Viewers can register themselves using their email addresses.

b. Login and Verification

- **Login:** Users will log in using their email and password.
- **Verification:** Upon registration, users will receive a verification email to confirm their accounts. Admins and content creators will have additional verification steps if required.

c. Users

- **Admin:** Has full control over platform operations, including user management, content moderation, and system settings.
- **Content Creators:** Manage their own channels, upload videos, and interact with their audience.
- **Moderators:** Oversee community interactions and content to ensure compliance with platform guidelines.
- **Viewers:** Access to watch videos, interact with content through comments, likes, and subscriptions.

d. Dashboards

- Each user role has access to a customized dashboard, displaying relevant controls and information based on their authorization level.

e. Manage Users

- Admins have the ability to assign, remove, or modify user roles such as content creators and moderators.

f. Manage Content

- Admins and content creators can upload, modify, or delete video content and manage channel information.

g. Manage Community Interaction

- Moderators and admins can oversee community features, including comments, likes, and community posts, to ensure a positive user experience.

4.3 Design Patterns

we identified categories of design patterns used in the development of complex software solution, namely:

1. Software design patterns, which constitute the optimal solutions employed by skilled object-oriented developers to general issues that software developers face during developments process.
2. Convenient design patterns that have proven valuable for constructing scalable, dependable and secure applications.

4.4 Database

MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas. As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a BSON (Binary JSON) format to store documents.

4.5 Data Structure

- A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.
- A data structure is not only used for organizing the data. It is also used for processing, retrieving, and storing data.
- There are different basic and advanced types of data structures that are used in almost every program or software system that has been developed. So, we must have good knowledge about data structures.

- **Registration**

DATA ITEM	DATA TYPE	DATA CONSTRAINT	DESCRIPTION
UserID: -	String	Not Null (Primary Key)	User unique id.
Username: -	String	Not Null, Unique	Username of user
Email: -	String	Not Null, Unique	Email id of user
Full_Name: -	String	Not Null	Full Name of user
Avatar: -	String	Not Null	Avatar image of user
Cover_Image: -	String		Cover Image of user behind the avatar
Watch_History: -	Array		Videos the user will watch will appear

Password: -	String	Not Null (check)	Login Password user
Refresh_Token	String		JWT token of user which will refresh the user's access token stored in the cookie

- **Login**

DATA ITEM	DATA TYPE	DATA CONSTRAINT	DESCRIPTION
Email/Username: -	String	Not Null (Primary Key)	User unique id.
Password: -	String	Not Null (check)	Login Password user

Test Case:

- Test Cases are frequently pre-defined series of instructions addressing the processes to be done to assess if the end output reflects the anticipated outcome.
- Predefined sets of inputs, conditions, and end results may be included in these instructions. However, in order to complete one's testing, one may wind up having an excessive number of test cases.
- To prevent such problems, one should identify the appropriate test cases design approach for their needs to cut the number of test cases significantly.

Test Case No	Test Case Description
Test 1: -	Check if a user enters an email or username, if not then it will show "Username or email is required."
Test 2: -	Check when a user enters valid email or username pop up login success! Another is showing User does not exist
Test 3: -	Check when a user enters valid Password pop up login success! Another is showing Please enter a valid password.

4.6 Security Issues

- Web application security is crucial to protecting data, customers, and organizations from data theft, interruptions in business continuity, or other harmful results of cybercrime.
- As it seemed that online video hosting app like this too can manipulated or being tampered due to validation issues, and low level of technology knowledge.
- API which is responsible for data binding is secured with JWT (JSON Web Tokens).
- JWT tokens will be introduced in order to maintain the security and uniqueness of sessions.
- As Discussed in GANTT CHART, we have a module called Authentication. According to that chart all the validations and verification part will be done at the end of backend, during database phase alongside.
- The JavaScript library “bcrypt” is used for the security of the users by hashing the passwords of the users so that if anyone found the password then he will find it in hashed format and will not be able to do anything.

4.7 Design Test

- After the specific aspects of each test were selected, acceptable limits were determined from the researched materials. Test procedures were created and revised. Data sheets were created to record the result of the tests.
- Good data sheets are vital to an effective procedure, as they guide the test technician through the process of properly executing the procedure.
- For each procedure the steps were determined with the specific results expected and safety of the technician in mind. After the procedure and to discover any error or areas for improvement.

4.8 Testing

- **Frontend:** React.js unit testing involves several key practices to ensure code quality. Developers working on the "VideoTube" project should consider the following tools
    - **Jest:** Jest is a popular testing framework for JavaScript, often used with React.js. It provides a comprehensive set of features including test runners, assertions, and mocks. Jest is easy to set up and is designed to work out of the box with minimal configuration, making it ideal for testing React components and logic.

      ```
      npm install --save-dev jest
      ```
    - **React Testing Library:** This is a testing utility that helps developers test React components more effectively by focusing on the way users interact with the components. It encourages testing practices that ensure your components work as expected in a real-world environment.

- **Backend:**
    - Automated Testing is crucial for ensuring the stability and performance of your web application. Automated tests allow you to quickly rerun tests during development, ensuring that your project meets quality standards.

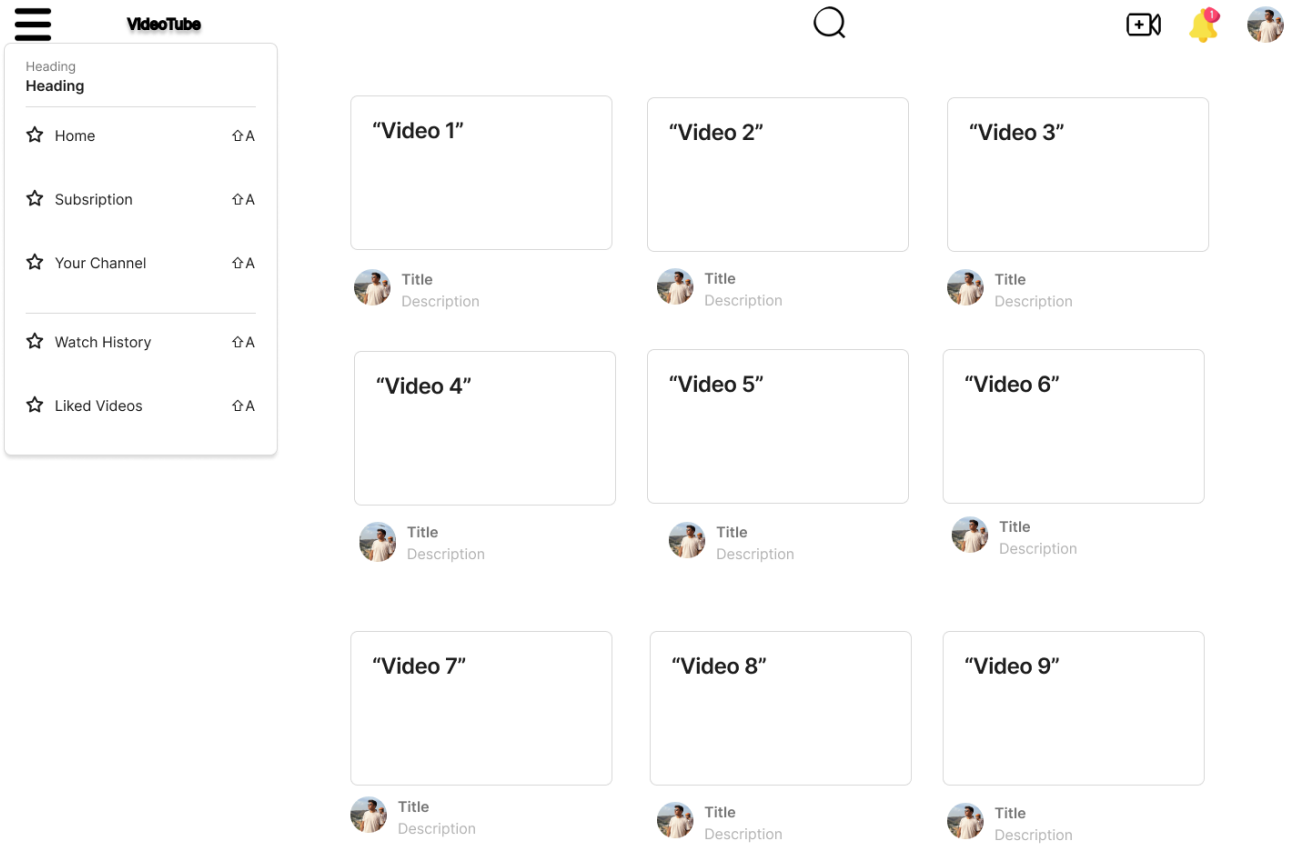
- Benefits of Automation include increased code coverage and faster feedback for developers. This enables you to detect and fix issues early in the development process, improving the overall reliability of your application.
- Productivity and Quality are both enhanced through automated testing. Tests can be run at key points in the development lifecycle, such as during code commits, feature integrations, and before releases, ensuring that the application remains stable and bug-free.
-

4.9 WIREFRAME

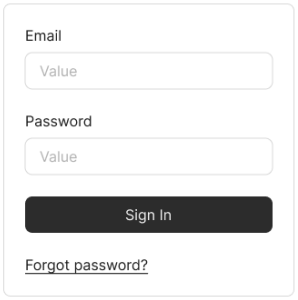
The designs you received are called wireframes (sometimes called wires, mockups, or mocks). A wireframe is a schematic, a blueprint, useful to help you and your programmers and designers think and communicate about the structure of the software or website you're building.

The same screen can be built in a lot of different ways, but only a few of them will get your message across correctly and result in an easy-to-use software or website. Nailing down a good interface structure is possibly the most important part of designing software.

a. Startup page



b. Login



C. Register